StorIQ

Serveur NAS

StorIQ System v. 8.0
Administration Avancée

Auteur: Emmanuel FLORAC
Réf. NAS-ADV-MAN
Version: 1.0.5

Mise à jour: 18/10/2021

Contacts :
tel: 01 78 94 84 00
support@intellique.com
info@Intellique.com

Intellique.com

Table des matières

Installation en mode avancé
Redimensionnement des partitions

parted ne sait pas redimensionner XFS. Il faut donc supprimer et recréer la partition avec exactement le même point de départ. C'est évidemment dangereux...

Si on agrandit le RAID (soit "tw_cli migrate", soit "arcconf MODIFY") depuis lequel on a démarré, le système ne peut pas toujours relire la table de partitions (sauf avec un contrôleur adaptec 7xxx ou supérieur). Dans ce cas il faut REDÉMARRER LE SERVEUR avant de tenter de manipuler les partitions. Après le redémarrage, on peut si nécessaire supprimer et recréer la partition agrandie.

Si on peut relire la table des partitions (utiliser `blockdev --rereadpt /dev/sdXX`); on doit redimensionner les pv et lv qui sont dessus, en utilisant `pvresize /dev/sdXX`, puis `lvextend -L +XXG /dev/vg/lv /dev/sdXX` puis utiliser *xfs_growfs* ou *tune2fs* pour retailler le système de fichiers.

ATTENTION Si on modifie la table de partition, ne pas oublier de faire `grub-install /dev/sda` (sur une StorIQ 3.x et supérieur).

1. Configuration des disques

Pour optimiser les performances, la règle générale est la suivante : minimiser l'antélecture (read-ahead) sur les disques physiques, le maximiser sur les disques logiques (RAID logiciel ou LVM). Une configuration typique comprenant un RAID logiciel avec deux disques /dev/sdb et /dev/sdc en RAID 0 LVM serait donc comme suit :

```
blockdev –setra 256 /dev/sdb
blockdev –setra 256 /dev/sdc
blockdev –setra 16384 /dev/vg0/lv0
```

La valeur optimale typique de read-ahead pour une machine avec des contrôleurs RAID matériels se situe entre 4096 et 131 072 selon le nombre et le type de disques. Une bonne approximation sera entre `1024 * <nombre de disques de données>` et `4096 * <nombre de disques de données>`.

Il faut aussi augmenter la longueur de file d'attente:

```
# 3Ware : 512
echo 512 > /sys/block/sda/queue/nr_requests

# Adaptec : 1024
echo 1024 > /sys/block/sda/queue/nr_requests
```

Attention, les Adaptec 5xxx ont un bug lors de fortes sollicitations elles cessent de répondre. Malheureusement il n'y a pas de correctif, on peut juste augmenter le *timeout* par défaut de 30 secondes à 45 (voire plus):

```
echo 45 > /sys/block/sda/device/timeout
```

On aura de bien meilleures performances en laissant le contrôleur RAID gérer lui-même les files d'attente d'IOs, donc on utilisera le scheduler "noop" sur les périphériques disques (c'est le réglage par défaut des noyaux StorIQ):

```
echo 'noop' > /sys/block/sda/queue/scheduler
```

Penser également à déclarer les SSD comme tels:

```
echo 0 > /sys/block/sda/queue/rotational
```

Selon le type de RAID et la vitesse de SSD, il peut être recommandé de désactiver le cache du contrôleur. La règle est la suivante:

- Pour du RAID-10 et des SSD rapides (SLC ou MLC), désactiver le cache en écriture et en lecture.
- Pour du RAID-5/6 et des SSD lents (TLC ou QLC), activer le cache en écriture et en lecture.

C'est naturellement une règle qui connaît des exceptions, à valider et contrôler en situation…

Optimisation des performances XFS

Pour optimiser les performances, on peut paramétrer le FS au moment du *mkfs* en fonction de la géométrie du raid (paramètres *swidtth* et *sunit*, ou *sw* et *su*). On doit aussi jouer sur les options de montage :

- *noatime* : pour minimiser les accès au journal
- *align*, *noalign* : pour que les limites de blocs du FS correspondent bien aux blocs du RAID
- *inode64* : pour les FS de plus de 4 To, permet de répartir les inodes au plus près des blocs de données. Rend incompatible avec les noyaux 32 bits et avec certains clients NFS, attention !
- *allocsize*=<*size*> : taille de préallocation (défaut 64Ko). S'il n'y a que des gros fichiers, mieux vaut mettre un *allocsize* grand voire très grand (1M ou plus).

Autres options :

Filestreams

Filestreams (sur XFS) permet d'allouer des fichiers successifs sur des *extents* contigus. Ainsi, sur des séries d'images numérotées, on assure que les fichiers seront optimalement regroupés à la lecture.

Avec un noyau récent, on peut utiliser "*filestreams*" soit en option de montage (*mount -o filestream*s) soit en activant l'attribut étendu *filestreams* sur un dossier pour indiquer que l'allocation doit se faire en mode "*filestreams*".

Autre option de montage classique, `logbufs=8,logbsize=256k` pour maximiser les buffers. Je n'ai pas remarqué d'effet sensible sur les systèmes modernes.

CoW (Copy on Write)

On peut faire des « snapshots » de fichiers ou de dossiers directement en faisant du *CoW* .

```
cp  --reflink vdisk.img vdisk_snap.img"
```

La taille de la section en *CoW* est définie par l'option "*cowextsize*" lors de la création du système de fichiers. Par défaut c'est 128Ko. Donc chaque modification de bloc de 4Ko entraîne un « snapshot » de 128Ko au maximum.

On peut utiliser *xfs_io* avec la commande "*extsize*" pour définir la taille *d'extent* demandée sur un dossier donné. Dans ce cas tous les fichiers du dossier hériteront de ce réglage. Il est donc possible d’optimiser la taille des *extents* en fonction de la taille des entrées/sorties, dossier par dossier.

Scrub

Lorsqu’on ajoute l'option *rmapbt=1* lors du mkfs il est possible d’utiliser le « scrubbing ». Il y a un impact sur la performance, selon Darrick J. Wong d’environ 10%.

Quotas XFS

XFS permet les quotas par dossier, (appelés "projets" en terminologie XFS)

- monter le FS avec l'option pquota ("project quota")
- créer un fichier /etc/projects listant les dossiers à utiliser avec quotas et leurs numéros (n'importe quoi mais unique) :

```
1:/mnt/raid/tamere
19:/mnt/raid/enslip
42:/mnt/raid/chezprisunic
```

- créer un fichier /etc/projid qui fait correspondre les numéros de projets avec un nom (label):

```
1:tamere
19:enslip
42:chezprisunic
```

- initialiser les quotas pour un projet en donnant le label ou le numéro (en pratique selon les version le numéro seul est sûr) :

```
xfs_quota -x -c "project -s 1"
```

- vérifier les quotas :

```
xfs_quota -x -c "project -c 1"
```

- status des quotas, se vérifie au niveau du FS :

```
~# xfs_quota -x -c report /mnt/raid
Project quota on /mnt/raid (/dev/md0)
                               Blocks
Project ID       Used       Soft       Hard    Warn/Grace
---------- --------------------------------------------------
1                   0          0          0     00 [--------]
```

Si on active les quotas dans le fstab et qu'on fait un `mount -o remount /mnt/truc`, l'option quota est bien listé dans la sortie de "mount" mais c'est un piège, les quotas ne sont pas actifs! de manière générale, beaucoup d'options de montage de XFS ne marchent pas lors d'un "remount", il faut donc démonter puis remonter.

Mise en veille des disques et XFS

Si on active la mise en veille des disques sur les contrôleurs RAID qui le supportent, il faut impérativement allonger le délai de time out du journal pour éviter des Ooops:

```
echo 720000 > /proc/sys/fs/xfs/xfssyncd_centisecs
```

Alternative : JFS

Dans certains cas rares JFS peut être mieux adapté. Son principal avantage est le support de 8192 ACLs par fichier, contre 25 seulement pour XFS. Ceci au prix d'une performance un peu inférieure à XFS, et l'impossibilité de défragmenter le système de fichiers -- souvent utile quand celui-ci est âgé.

2.Configuration du cache

On peut utiliser soit **bcache**, soit **LVM cache**. Pour des raisons de simplicité de configuration seule cette dernière option sera décrite ici.

LVM Cache utilise deux volumes logiques : un volume de métadonnées et un volume de cache. De plus, il réserve un espace de secours équivalent à la taille du volume de métadonnées, afin de pouvoir faire un cliché en cas de corruption.

Le volume “métadonnées” doit faire au moins 1/1000e du volume de cache.

Le volume de cache, de métadonnées et le volume à cacher doivent être dans le même groupe de volumes.

Dans les exemples suivants, */dev/sda* est le SSD, */dev/sdb* le disque dur. */dev/vg0* est le groupe de volume. */dev/sda5* et */dev/sdb5* sont des PV du VG *vg0* :

```
# pvs
  PV         VG   Fmt  Attr PSize   PFree
  /dev/sda5  vg0  lvm2 a--  199,21g    0
  /dev/sdb5  vg0  lvm2 a--    4,52t    0
# vgs
  VG   #PV #LV #SN Attr   VSize VFree
  vg0    2   2   0 wz--n- 4,72t    0
```

On crée le volume de métadonnées et le volume de cache en spécifiant le volume physique à utiliser (le SSD). Le volume de métadonnées doit faire au moins 1/1000 de la taille du volume de cache:

```
# lvcreate -L 250M -n CacheMeta vg0 /dev/sda5

# lvcreate -l 100%PVS -n Cache vg0 /dev/sda5
```

Attention, il est préférable de créer le cache avant le volume à cacher. Ainsi un volume caché sera créé automatiquement pour les clichés des métadonnées, sans avoir à réserver l'espace.

```
# lvconvert --type cache-pool --poolmetadata vg0/CacheMeta vg0/Cache
  WARNING: Converting logical volume vg0/Cache and vg0/CacheMeta to pool's
data and metadata volumes.
  THIS WILL DESTROY CONTENT OF LOGICAL VOLUME (filesystem etc.)
Do you really want to convert vg0/Cache and vg0/CacheMeta? [y/n]: y
  Logical volume "lvol0" created
  Converted vg0/Cache to cache pool.
```

Ensuite on crée le volume à cacher:

```
# lvcreate -l100%PVS -n raid vg0 /dev/sdb5
  Logical volume "raid" created
```

Puis on le déclare caché:

```
# lvconvert --type cache --cachepool vg0/Cache vg0/raid
  Logical volume vg0/raid is now cached.
```

Et voilà :

```
# lvs
  LV    VG   Attr       LSize   Pool  Origin       Data%  Meta%  Move Log Cpy%Sync Convert
  Cache vg0  Cwi---C--- 198,96g
  raid  vg0  Cwi-a-C---   4,52t Cache [raid_corig]
```

lvs -a montre les volumes "cachés" de secours :

```
# lvs -a
  LV              VG   Attr       LSize   Pool  Origin       Data%  Meta%  Move Log Cpy%Sync Convert
  Cache           vg0  Cwi---C--- 198,96g
  [Cache_cdata]   vg0  Cwi-ao---- 198,96g
  [Cache_cmeta]   vg0  ewi-ao---- 252,00m
  [lvol0_pmspare] vg0  ewi------- 252,00m
  raid            vg0  Cwi-a-C---   4,52t Cache [raid_corig]
  [raid_corig]    vg0  owi-aoC---   4,52t
```

3.Chiffrement des partitions

créer un filesystem crypté

On installe d'abord cryptsetup, puis : Par exemple pour crypter /dev/md0 en AES 256 :

```
cryptsetup -c aes-cbc-essiv:sha256 -y -s 256 luksFormat /dev/md0
```

Pour utiliser le périphérique :

```
cryptsetup luksOpen /dev/md0 crypt1
```

Ensuite il faut penser à créer un filesystem et modifier le /etc/fstab :

```
mkfs -t xfs /dev/mapper/crypt1
```

Enfin il faut remplir le fichier /etc/crypttab avec les informations nécessaires. Voir le "man crypttab". On peut soit utiliser un mot de passe soit par exemple une clef USB, ou autre.

accès par mot de passe

Exemple de crypttab correspondant:

```
crypt1  /dev/md0        none    tries=3,timeout=60,loud,luks
```

Utiliser une clé USB pour stocker la clef de chiffrement

Pour crypter en utilisant un fichier sur une clef USB, le mieux est de générer un fichier aléatoire avec /dev/random, un une clef avec ssl-keygen, gpg, etc. Ensuite on met ce fichier sur une clef USB avec un point de montage et on l'utilise pour verrouiller le volume :

```
 cryptsetup luksAddKey /dev/md0 /mnt/key/crypt1.key
Enter any LUKS passphrase:
key slot 0 unlocked.
Command successful.
```

Ensuite il faut modifier le crypttab pour utiliser cette clef plutôt qu'un prompt, ainsi :

```
crypt1  /dev/md0        /mnt/key/crypt1.key     loud,luks
```

Là où ça se corse, c'est qu'il faut qu'au démarrage, la clef USB soit pleinement activée _avant_ le démarrage de /etc/init.d/cryptdisks, ensuite il faut que le montage des FS locaux soit fait suffisamment tard. J'ai modifié la séquence de démarrage ainsi :

```
cryptdisks              :   48              13   13   13   48
cryptdisks-early        :   59              11   11   11   59
mountall.sh             :                   15   15   15   35
```

J'ai donc décalé le démarrage normal de cryptdisks dans les runlevels 2,3,4, puis j'ai refait un mountall après. Par ailleurs j'ai modifié le /etc/default/cryptdisks pour qu'il monte la clef tout seul:

```
~# more /etc/default/cryptdisks
# Run cryptdisks at startup ?
CRYPTDISKS_ENABLE=Yes

# Mountpoints to mount, before starting cryptsetup. This is useful for
# keyfiles on removable media. Seperate mountpoints by space.
CRYPTDISKS_MOUNT="/mnt/key"

# Default check script, see /lib/cryptsetup/checks/
# Takes effect, if the 'check' option is set in crypttab without a value
CRYPTDISKS_CHECK=vol_id

# Default precheck script, see
# Takes effect, if the 'precheck' option is set in crypttab without a value
CRYPTDISKS_PRECHECK=

# Default timeout in seconds for password prompt
# Takes effect, if the 'timeout' option is set in crypttab without a value
CRYPTDISKS_TIMEOUT=180
```

Voici le point de montage de la clef USB (dans /etc/fstab) :

```
/dev/disk/by-id/usb-JUNGSOFT_NEXDISK-part1 /mnt/key vfat
noauto,rw,uid=0,gid=0,umask=277 0 0
```

Noter qu'elle est en noauto, comme ça elle est montée et démontée au démarrage de diskcrypt, on peut donc l'insérer avant le boot, attendre la fin du boot et la récupérer.

4.Configuration réseau

10 Gigabits

Myricom propose plein d'optimisations. Voir (le site)https://www.myricom.com/scs/READMES/README.myri10ge-linux. On retiendra surtout d'ajouter à /etc/sysctl.conf:

```
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.core.netdev_max_backlog = 250000
```

Agrégation des ports réseaux

Paramétrages bonding : le 802.3ad nécessite 2 machines configurées sur le réseau ou une configuration du switch. Il n'y a pas d'amélioration des performances si le switch ne coopère pas (c'est à dire jamais...) Balance-alb ou balance-rr sont les meilleurs en performances mais cela varie selon les protocoles, les switchs, le nombre d'interface réseau... Il faut tester chaque configuration. En pratique balance-tlb est le plus "tranquille" : gain de performance et pas de problème de compatibilité.

Il est possible de configurer plusieurs groupes d'interfaces 'bond0, bond1, etc.) avec des modes de fonctionnement distincts. Voir la documentation de « bonding_cli ».

Authentification LDAP

Commencer par faire un

```
dpkg-reconfigure libpam-ldap
```

Ensuite modifier */etc/nsswitch.conf* ainsi :

```
passwd:     files ldap compat
group:      files ldap compat
shadow:     files ldap compat
```

Puis modifier les fichiers dans /etc/pam.d suivants : */etc/pam.d/common-account :*

```
account sufficient    pam_ldap.so
account required    pam_unix.so try_first_pass
```

/etc/pam.d/common-auth :

```
auth sufficient        pam_ldap.so
auth required        pam_unix.so nullok_secure try_first_pass
```

/etc/pam.d/common-password :

```
password sufficient    pam_ldap.so
password required    pam_unix.so nullok obscure min=4 max=8 md5
try_first_pass
```

Ensuite il faut entrer les paramètres du serveur LDAP dans */etc/ldap/ldap.conf :*

```
BASE  dc=example,dc=org
URI   ldap://192.168.1.30
```

Pour finir, on redémarre nscd :

```
service nscd restart
```

5.Partage de fichiers

NFS

Note : Pour les clients Macs et BSD, il faut exporter en mode "*insecure*".

J'ai fabriqué un petit moniteur des serveurs NFS (2,3 et 4), *munin-nfsd-perf*. Il va automatiquement se configurer. Il donne 4 valeurs :

- *CPU time %* : le pourcentage de CPU consommé par les démons NFS.
- *requests/s* : le nombre de connexions par secondes.
- *working threads %* : le pourcentage de démons nfsd qui travaillent le plus.
- *idle threads %* : le pourcentage de démons nfsd qui ne font presque rien.

La somme des "*workers*" et des "*idlers*" est normalement inférieure à 100%. S'il y a une forte proportion de "*idlers*" (plus de 10 ou 20%), c'est qu'il y a plus de démons *nfsd* qu'il n'est nécessaire (abaisser la valeur de *RPCNFSDCOUNT* dans */etc/default/nfs-kernel-server* et relancer). S'il y a une forte proportion de "*workers*" (plus de 50%), il peut ne pas y avoir assez de démons *nfsd*. De même si le nombre de connexions par seconde est supérieur au nombre de clients connectés, cela signifie que les clients « migrent » sans cesse d'un serveur à l'autre, donc qu'il n'y a sans doute pas suffisamment de démons.

Problème ennuyeux le plus courant avec NFS: le message "*rpc.statd not responding, timed out*". En général la procédure suivante résout le problème :

```
service nfs stop

service rpcbind stop

rm -rf /var/lib/nfs/statd/sm/*

rm -rf /var/lib/nfs/statd/sm.bak/*

service rpcbind start

service nfs start
```

NFS4

NFS4 utilise le concept de "racine virtuelle unique". Les exports font partie d'un pseudo-fs, identifiés par un "fsid" (toujours 0, pour l'instant). *Tous les exports* doivent être des sous-dossiers de la racine virtuelle. Pour les détails : https://wiki.linux-nfs.org/wiki/index.php/Nfsv4_configuration_fr

SAMBA

Samba et les Macs

Les performances sont très faibles comparées à NFS.

Le finder de Mac OS X 10.5 ne peut pas (via command-K) se connecter à un partage samba si le nom du serveur contient un tiret (-). On arrive à forcer en utilisant comme nom d'utilisateur `<adresse IP>\user` cependant.

Si on utilise des droits un tant soit peu sophistiqués dans un réseau mixte Windows/Mac, il faut désactiver les extensions Unix dans Samba, sinon ça pose des problèmes d'accès. Ajouter dans la section "global" de *smb.conf*:

```
unix extensions = no
```

puis redémarrer samba.

Authentification SAMBA avec LDAP

Ajouter la section suivante dans /etc/samba/smb.conf :

```
;LDAP-specific settings
   ldap admin dn = "cn=Manager,dc=syroidmanor,dc=com"
   ldap server = localhost
   ldap port = 389
   ldap ssl = no
   ldap suffix = "ou=Users,dc=syroidmanor,dc=com"
```

En remplaçant naturellement les informations serveur données en exemple. Par contre pour être complet il faut utiliser les scripts IdealX de synchronisation des mots de passe avec le LDAP.

Configuration Samba 4 avec AD Windows 2012 et supérieurs

S'assurer qu'on a installé tous les paquets :

```
apt-get install samba acl attr quota fam winbind libpam-winbind libpam-krb5
libnss-winbind krb5-config krb5-user ntp dnsutils ldb-tools
```

Arrêter les services :

```
service smbd stop
service nmbd stop
service winbind stop
```

Exemple de */etc/samba/smb.conf :*

```
[global]
    workgroup = example
    security = ADS
    realm = EXAMPLE.LAN

    dedicated keytab file = /etc/krb5.keytab
    kerberos method = secrets and keytab
    server string = Data %h

    winbind use default domain = yes
    winbind expand groups = 4
    winbind nss info = rfc2307
```

```
    winbind refresh tickets = Yes
    winbind offline logon = yes
    winbind normalize names = Yes

    ## map ids outside of domain to tdb files.
    idmap config *:backend = tdb
    idmap config *:range = 2000-9999
    ## map ids from the domain  the ranges may not overlap !
    idmap config EXAMPLE : backend = rid
    idmap config EXAMPLE : range = 10000-999999
    template shell = /bin/bash
    template homedir = /home/EXAMPLE/%U

    domain master = no
    local master = no
    preferred master = no
    os level = 20
    map to guest = bad user
    host msdfs = no

    # user Administrator workaround, without it you are unable to set privileges
    username map = /etc/samba/user.map

    # For ACL support on domain member
    vfs objects = acl_xattr
    map acl inherit = Yes
    store dos attributes = Yes

    # Share Setting Globally
    unix extensions = no
    reset on zero vc = yes
    veto files = /.bash_logout/.bash_profile/.bash_history/.bashrc/
    hide unreadable = yes

    # disable printing completely
    load printers = no
    printing = bsd
    printcap name = /dev/null
    disable spoolss = yes
```

Créer le fichier */etc/samba/user.map* contenant *:*

```
!root = EXAMPLE\Administrateur EXAMPLE\administrateur Administrateur
administrateur
```

Mettre dans */etc/krb5.conf :*

```
[libdefaults]
    default_realm = EXAMPLE.LAN
    dns_lookup_realm = false
    dns_lookup_kdc = true
```

Ensuite exécuter les commandes suivantes :

```
chmod 644 /etc/krb5.conf

net ads join -U Administrateur
Using short domain name -- EXAMPLE
Joined 'DEBMEMBER' to dns domain 'EXAMPLE.lan'

service smbd start
service nmbd start
service winbind start
```

Éditer */etc/nsswitch.conf* en ajoutant '*winbind*' aux lignes '*passwd*' et *'group'* de façon à ce que la commante *getent passwd* retourne tous les utilisateurs, locaux et de l'AD.

Enfin tester la configuration :

```
# wbinfo -u
rowland
test
storiq
# getent passwd rowland
rowland:*:11107:10513:Rowland Penny:/home/rowland:/bin/bash
```

Désactivation complète de SMBv1

Modifier */etc/samba/smb.conf* en rajoutant dans la section *[global]* :

```
client ipc min protocol = SMB2
client min protocol = SMB2
server min protocol = SMB2
```

On peut vérifier si SMB v1 est actif ou non avec *smbclient,* en forçant la version du protocole d'authentification en « NT1 ». Example de commande à adapter :

```
# smbclient //localhost/partage -U storiq -m NT1
```

interface avancée de partage réseau Windows

Vous pouvez modifier les paramètres de création des UID et GID des utilisateurs du Domaine. Cependant ce n'est utile que si vous utilisez un autre serveur Samba (Unix, Linux, Mac OS X...) ou un autre NAS (NetApp...) et que vous souhaitez partager les données d'authentification avec ces derniers (par exemple pour des partages NFS).

Netatalk

La version 3 de *netatalk* utilise des formats de configuration différent des versions précédentes (bouuuh), proche de celui de samba (aaah). Chaque fichier et dossier dans un partage AFP reçoit un identifiant 32 bits, qui est stocké dans une base CNID. Les bases CNID peuvent devenir assez grosses (dans */var/netatalk/CNID*). Il peut être nécessaire sur les gros partages avec beaucoup d'activité de déplacer ce dossier afin qu'il ait suffisamment de place pour s'étendre.

On peut régénérer la base CNID sans avoir arrêté le service *netatalk* avec la commande

```
dbd -f <chemin du partage>
```

Ça peut être assez long et demander beaucoup de ressources, il faut bien sûr le lancer la nuit uniquement.

C'est le démon *cnid_metad* qui gère ces bases, en cas de problèmes il faut s'assurer qu'il tourne bien.

VSFTPd

Les vidéastes appellent très improprement "FTP passif" le FXP. Pour activer le mode FXP dans VSFTPd il faut mettre ceci dans le /etc/vsftpd.conf:

```
pasv_promiscuous=yes
port_promiscuous=yes
```

iSCSI

Configuration de la cible avec target-cli

- installer *targetcli*
- lancer en *root* la commande *targetcli*

```
Warning: Could not load preferences file /root/.targetcli/prefs.bin.
targetcli shell version 2.1.fb48
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.
```

La commande "*ls*" liste les objets :

```
/> ls
o- / ..........................................................[...]
  o- backstores ...............................................
[...]
  | o- block ..................................... [Storage Objects: 0]
  | o- fileio .................................... [Storage Objects: 0]
  | o- pscsi ..................................... [Storage Objects: 0]
  | o- ramdisk ................................... [Storage Objects: 0]
  o- iscsi ............................................... [Targets: 0]
  o- loopback ............................................ [Targets: 0]
  o- vhost ............................................... [Targets: 0]
  o- xen-pvscsi .......................................... [Targets: 0]
/>
```

Les différents items :

« *Backstores* » correspond aux ressources de stockage:

- Les objets "*block*" sont des périphériques bloc, typiquement des volumes logiques LVM (ou des partitions, des disques, etc).
- Les objets "*fileio*" sont des fichiers sur disque.
- Les objets "*pscsi*" sont du "*passthrough*" SCSI. Cela permet par exemple d'exporter un lecteur de bande, de CD, etc.
- Les objets "*ramdisk*" sont des disques mémoires virtuels (non persistants !)

Ensuite nous avons les différents modes d'exportation de ces différentes ressources:

- *iscsi* (cible iscsi)
- *loopback* (local uniquement)

- *vhost* (pour les machines virtuelles KVM)
- *xen-pvscsi* (pour les machines virtuelles Xen)

Il existe aussi d'autres modes selon les logiciels et matériels installés, comme *qla2xxx* (export sur Fibre Channel), etc.

Création d'une ressource de stockage

Type bloc :

```
cd /backstores/blocks
create name=lvm_vol1 dev=/dev/vg0/iscsilv1
Created block storage object lvm_vol1 using /dev/vg0/iscsilv1.
```

Type fichier :

```
cd /backstores/fileio
/backstores/fileio> create name=file_vol1
file_or_dev=/var/lib/iscsi/file_vol1.img size=2G
Created fileio file_vol1 with size 2147483648
```

Affichons l'état actuel de nos objets :

```
cd /
ls
o- / ..................................................................................... [...]
  o- backstores .......................................................................... [...]
  | o- block .............................................................. [Storage Objects: 1]
  | | o- lvm_vol1 ........................ [/dev/vg0/iscsilv1 (5.0GiB) write-thru deactivated]
  | |   o- alua ................................................................ [ALUA Groups: 1]
  | |     o- default_tg_pt_gp ................................ [ALUA state: Active/optimized]
  | o- fileio ............................................................. [Storage Objects: 1]
  | | o- file_vol1 ..............[/var/lib/iscsi/file_vol1.img (2.0GiB) write-back deactivated]
  | |   o- alua ................................................................ [ALUA Groups: 1]
  | |     o- default_tg_pt_gp ................................ [ALUA state: Active/optimized]
  | o- pscsi .............................................................. [Storage Objects: 0]
  | o- ramdisk ............................................................ [Storage Objects: 0]
  o- iscsi ........................................................................ [Targets: 0]
  o- loopback ..................................................................... [Targets: 0]
  o- vhost ........................................................................ [Targets: 0]
  o- xen-pvscsi ................................................................... [Targets: 0]
/>
```

N'oublions pas de sauver la configuration :

```
saveconfig
```

Création d'une cible iSCSI

On crée une cible avec la commande "*create*", optionnellement suivie d'un nom de cible, exemple :

```
create iqn.2003-03.com.intellique.srv02:tgt1
```

Si on ne donne pas de nom, un nom aléatoire est créé :

```
cd /iscsi
/iscsi> create
Created target iqn.2003-01.org.linux-iscsi.debian10.x8664:sn.539385af292e.
Created TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260.
```

Une cible iSCSI est annoncée par un portail. Nous avons vu qu'un portail par défaut écoutant sur toutes les interfaces a été créé. On peut souhaiter n'écouter que sur certaines IP, auquel cas il faut supprimer le portail par défaut :

```
cd /iscsi/<iqn...>/tpg1/portals
/iscsi/iqn.20.../tpg1/portals> delete 0.0.0.0 3260
Deleted network portal 0.0.0.0:3260
```

On peut ensuite en créer d'autres:

```
/iscsi/iqn.20.../tpg1/portals> create 10.0.4.224 3260
Using default IP port 3260
Created network portal 10.0.4.224:3260.
/iscsi/iqn.20.../tpg1/portals> ls
o- portals ..................................................... [Portals: 1]
  o- 10.0.4.224:3260 ..................................................... [OK]
```

Association d'une cible avec un volume de stockage

Nous pouvons lier les volumes créés précédemment avec notre cible comme ceci :

```
cd /iscsi/iqn.20...92e/tpg1/luns
/iscsi/iqn.20...92e/tpg1/luns> create /backstores/block/lvm_vol1
Created LUN 0.

/iscsi/iqn.20...92e/tpg1/luns> create /backstores/fileio/file_vol1
Created LUN 1.
```

Enfin, sauvons la configuration et regardons ce que ça donne :

```
> cd /
/> saveconfig
Last 10 configs saved in /etc/rtslib-fb-target/backup.
Configuration saved to /etc/rtslib-fb-target/saveconfig.json
/> ls
o- / ..................................................................... [...]
  o- backstores ..................................................... [...]
  | o- block ..................................... [Storage Objects: 1]
  | | o- lvm_vol1 ..................... [/dev/vg0/iscsilv1 (5.0GiB) write-thru activated]
  | |   o- alua ....................................... [ALUA Groups: 1]
  | |     o- default_tg_pt_gp ........................... [ALUA state: Active/optimized]
  | o- fileio ..................................... [Storage Objects: 1]
  | | o- file_vol1 .............[/var/lib/iscsi/file_vol1.img (2.0GiB) write-back activated]
  | |   o- alua ....................................... [ALUA Groups: 1]
  | |     o- default_tg_pt_gp ........................... [ALUA state: Active/optimized]
```

```
  | o- pscsi ..................................................... [Storage Objects: 0]
  | o- ramdisk ................................................... [Storage Objects: 0]
  o- iscsi ..................................................................... [Targets: 1]
  | o- iqn.2003-01.org.linux-iscsi.debian10.x8664:sn.539385af292e ................ [TPGs: 1]
  |   o- tpg1 ....................................................... [no-gen-acls, no-auth]
  |     o- acls ................................................................... [ACLs: 0]
  |     o- luns ................................................................... [LUNs: 2]
  |     | o- lun0 .................. [block/lvm_vol1 (/dev/vg0/iscsilv1) (default_tg_pt_gp)]
  |     | o- lun1 ...... [fileio/file_vol1 (/var/lib/iscsi/file_vol1.img) (default_tg_pt_gp)]
  |     o- portals ............................................................. [Portals: 1]
  |       o- 10.0.4.224:3260 ..................................................... [OK]
  o- loopback .................................................................. [Targets: 0]
  o- vhost ..................................................................... [Targets: 0]
  o- xen-pvscsi ................................................................ [Targets: 0]
/>
```

Gestion des droits d'accès

Authentification par initiateur :

```
cd /iscsi/iqn.20...85af292e/tpg1

/iscsi/iqn.20...85af292e/tpg1> set attribute authentication=0
Parameter authentication is now '0'.
```

Pour créer une autorisation pour un initiateur défini:

```
cd /iscsi/iqn.20...92e/tpg1/acls
/iscsi/iqn.20...92e/tpg1/acls> create wwn=iqn.1993-
08.org.debian:01:02ca36d9db
Created Node ACL for iqn.1993-08.org.debian:01:02ca36d9db
Created mapped LUN 1.
Created mapped LUN 0.
```

À partir de là, l'initiateur défini peut se connecter sans authentification.

Gestion de l'authentification

On doit d'abord activer l'authentification :

```
/iscsi/iqn.20...85af292e/tpg1> set attribute authentication=1
Parameter authentication is now '1'.
```

Ensuite on peut créer une connexion par utilisateur/mot de passe :

```
cd /iscsi/iqn.20...85af292e/tpg1>
/iscsi/iqn.20...85af292e/tpg1> set auth userid=user
Parameter userid is now 'usr'.
/iscsi/iqn.20...85af292e/tpg1> set auth userid=bolo42
Parameter password is now 'bolo42'.
```

Configuration de l’initiateur Linux

Avant toute chose il faut démarrer le service *open-iscsi* :

```
service open-iscsi start
```

Il faut d'abord « découvrir » les cibles :

```
 iscsiadm -m discovery  --type sendtargets --portal 192.168.2.23
```

Ensuite on peut lister les nœuds ainsi découverts :

```
iscsiadm -m node
```

ensuite on fait une connexion (« login ») :

```
iscsiadm -m node -T  iqn.2003-03.com.intellique:contrebasse.test.iscsi1 -l
```

Enfin pour se déconnecter (« logout ») :

```
iscsiadm -m node -T  iqn.2003-03.com.intellique:contrebasse.test.iscsi1 -p 192.168.2.23:3260 -u
```

Pour démarrer le montage des cibles automatiquement :

```
iscsiadm -m node -T <Storage IQN 1> -p <Storage IP 1>:3260 -o update -n node.conn[0].startup -v automatic
iscsiadm -m node -T <Storage IQN 2> -p <Storage IP 1>:3260 -o update -n node.conn[0].startup -v automatic
```

Pour forcer la connexion automatique sur toutes les cibles, ajouter dans */etc/iscsid.conf* :

```
node.startup = Automatic
```

Initiateur pour MacOS

Nous avons testé *GlobalSAN iSCSI Initiator* de *Studio Network Solutions*. C'est un initiator gratuit pour MacOS. On peut le télécharger ici :

https://www.snsftp.com/public/globalsan/

La configuration est aisée.

Optimisation côté initiateur (client)

Attention à l'alignement des volumes. Par défaut, la table de partitions MS-DOS introduit un décalage : quand on crée une table de partition sur un périphérique iSCSI, on peut décaler les blocs par rapport au périphérique sous-jacent, ce qui fait que chaque lecture ou écriture induit deux lectures ou deux écritures sur le disque physique, très mauvais pour les performances !

Pour aligner les volumes :

1. Avec un initiateur Linux, ne pas créer de table de partition sur un périphérique iSCSI est encore le mieux.
2. avec un initiateur Windows, on est obligé de créer une table de partitions. Utiliser la ligne de commande DISKPART :

```
C:\>diskpart
DISKPART> list disk
DISKPART> select disk 1
DISKPART> list partitions
DISKPART> create partition primary align=64
```

Il faut que l'offset (ici 64K, soit 65536) soit toujours un multiple de 4096, ou idéalement de la taille de *stripe* du périphérique physique utilisé.

Optimisation côté cible (serveur)

Pour optimiser pour les IOPS, partir de cette configuration:

```
Target iqn.2012-06.test5:testing
        Lun 0 Path=/dev/md100,Type=blockio
        MaxConnections          8
        InitialR2T              No
        ImmediateData           Yes
        MaxRecvDataSegmentLength 65536
        MaxXmitDataSegmentLength 65536
        MaxBurstLength          1048576
        FirstBurstLength        262144
        MaxOutstandingR2T       1
        HeaderDigest            None
        DataDigest              None
        NOPInterval             60
        NOPTimeout              180
        Wthreads                8
        QueuedCommands          64
```

iSCSI et VMWare

VMware a besoin des SCSI ID et SN pour identifier correctement les LUNs des différentes cibles. Vous pouvez forcer des valeurs manuelles par exemple si vous souhaitez mettre les cibles iSCSI en cluster, etc.

En cas de problème de déconnexion sous forte charge, il faut modifier les paramètres VMware. On peut faire ceci sur le serveur VM:

```
esxcfg-advcfg -s 14000 /VMFS3/HBTokenTimeout
```

Autre possibilité, modifier dans les paramètres avancés de l'initiateur iSCSI de VMware :

```
MaxCommands             1
```

Serveur de courrier électronique intégré

Le serveur est exim4. Utiliser **dpkg-reconfigure exim4-config**.

6.Mise en grappe des serveurs

OCFS2

Il suffit de faire

```
aptitude install ocfs2console
```

pour tout installer.

D'abord, configurer avec `dpkg-reconfigure ocfs2-tools .`

Tres important : il faut beaucoup augmenter le délai *heartbeat* "seuil de battement O2CB" : par défaut il est à 7; normalement une bonne valeur se situe entre 30 et 35.

Ne pas oublier de faire *service o2cb enable*.

Ensuite, configurer le cluster avec *ocfs2console* : d'abord configurer les nœuds (*cluster -> configure nodes*), ensuite configurer les systèmes de fichiers en les formatant si nécessaire. Faire le montage depuis ocfs2console, puis copier les lignes correspondantes de `/etc/mtab` dans `/etc/fstab` sur tous les nœuds.

L'erreur la plus courante avec OCFS2 c'est le time-out, dans ce cas il se peut que le OCFS2 *heartbeat* soit regle trop court sur une machine, verifier (`/etc/default/o2cb`), l'autre erreur classique etant le "out of memory" : avec 1Go par CPU c'est juste, très juste.

On peut agrandir un volume OCFS2 à la volée après l'avoir démonté sur tous les nœuds: on redimensionne la partition (avec *parted* ou autre) puis on agrandit le volume avec *tunefs.ocfs2 -S /dev/sdXX*. Ensuite on relit la table de partition sur tous les nœuds avec *blockdev --rereadpt /dev/sdXX*, après quoi on peut remonter le volume.

Cluster de haute disponibilité avec DRBD et Corosync

Réplication DRBD
configuration de base

Les *drbd tools* contiennent un fichier de conf d'exemple : */etc/drbd.d/global_common.conf*, utilisant **drbd** comme nom de ressource et */dev/drbd0* comme périphérique commun.

1. Modifier ce fichier pour définir les périphériques et les hôtes à synchroniser.
2. Modifier /etc/hosts pour que les machines soient bien identifiées
3. S'assurer que les machines ont leurs horloges bien synchronisées, si possible utiliser un serveur de temps pour les deux.
4. supprimer éventuellement la ligne `include "drbd.d/*.res";` dans */etc/drbd.conf*

initialisation

1. démarrer le service sur les nœuds.
2. Vérifier le statut actif/passif dans */proc/drbd*

Si nécessaire, créer la signature :

```
drbdadm  create-md drbd
```

Lancer la synchronisation:

```
drbdadm -- --overwrite-data-of-peer primary drbd
```

Vérifier l'avancement avec `cat /proc/drbd`

Mode actif/actif

passer chaque noeud en primary avec

```
drbdadm primary all
```

vérifier le statut avec `cat /proc/drbd`

Utiliser le périphérique /dev/drbd0 normalement. Pour l'utiliser simultanément sur les deux noeuds, il faut employer OCFS2 (ou un autre système de fichier « cluster »).

Démarrer un nœud seul

Très utile, par exemple si on prévoit de passer plus tard à un cluster, mais qu'on n'a pas encore installé le second nœud.

```
drbdadm up res0
drbdadm primary res0 --force
```

ajouter un nœud à un cluster déjà existant

Copier la configuration depuis le premier nœud. Créer le périphérique :

```
drbdadm  create-md rd0
```

démarrer le service :

```
drbdadm up rd0
```

passer en secondaire :

```
drbdadm secondary rd0
```

attendre la fin de la synchronisation.

```
# cat /proc/drbd
version: 8.4.5 (api:1/proto:86-101)
srcversion: EDE19BAA3D4D4A0BEFD8CDE
 0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r---n-
    ns:1203334 nr:0 dw:25821 dr:1347863 al:2 bm:0 lo:0 pe:2 ua:70 ap:0 ep:1
wo:d oos:5582604
        [>...................] sync'ed:  7.5% (5448/5880)M
        finish: 0:09:15 speed: 10,044 (9,804) K/sec
```

Il peut être nécessaire de forcer la synchronisation sur le primaire :

```
~# drbdadm -- --overwrite-data-of-peer primary all
~# service drbd restart
```

reconnecter après une bascule

sur le secondaire:

```
drbdadm secondary rd0
drbdadm disconnect rd0
drbdadm -- --discard-my-data connect rd0
```

sur le primaire:

```
drbdadm connect rd0
```

Configuration Corosync

On utilisera soit DRBD, soit OCFS2, soit les deux à la fois.

7.Installation de corosync

Installer *pacemaker* et *crmsh :*

```
aptitude update
aptitude install -t pacemaker crmsh
```

Modifier */etc/hosts* pour que les deux nœuds se reconnaissent correctement. Ici on utilise les noms "cl1" et "cl2".

configurer *corosync* sur le premier nœud en mettant dans *corosync.conf :*

```
totem {
    version: 2
    secauth: off
    cluster_name: pacemaker1
    transport: udpu
}

nodelist {
    node {
        ring0_addr: cl1
        nodeid: 101
    }
    node {
        ring0_addr: cl2
        nodeid: 102
    }
}

quorum {
   provider: corosync_votequorum
   two_node: 1
   wait_for_all: 1
   last_man_standing: 1
   auto_tie_breaker: 0
}
```

Générer les clefs de chiffrement :

```
corosync-keygen
```

copier *corosync.conf* et *authkey* vers l'autre nœud. Redémarrer le service sur les deux nœuds.

Vérifier que tout fonctionne avec `crm status` .

Configurer avec crm

crm est un shell. Il est probablement judicieux de se familiariser avec…

```
crm configure
```

Exemple de configuration avec iSCSI:

```
primitive drbd_res ocf:linbit:drbd \
        params drbd_resource=rd0 \
        op monitor interval=29s role=Master \
        op monitor interval=31s role=Slave
primitive fs_res Filesystem \
        params directory="/mnt/raid" device="/dev/drbd0" fstype=xfs \
        meta target-role=Started
primitive iscsi_res lsb:iscsi-target \
        op monitor interval=10s role=Started \
        meta target-role=Started
primitive net_res ocf:redhat:ip.sh \
        params address=192.168.11.240 prefer_interface=bond0 \
        meta is-managed=true \
        meta target-role=Started
ms drbd_master_slave drbd_res \
        meta globally-unique=false clone-max=2 notify=true target-
role=Started
colocation iscsi_colo inf: iscsi_res net_res fs_res drbd_master_slave:Master
order iscsi_order Mandatory: drbd_master_slave:promote fs_res:start
net_res:start iscsi_res:start
location master-on-NAS1 drbd_master_slave \
        rule $role=master 100: #uname eq NAS1
property stonith-enabled=off
```

Avec NFS (marche parfaitement en NFS3, moins en NFS4):

```
primitive drbd_res ocf:linbit:drbd \
        params drbd_resource=rd0 \
        op monitor interval=29s role=Master \
        op monitor interval=31s role=Slave
primitive fs_res Filesystem \
        params directory="/mnt/raid" device="/dev/drbd0" fstype=xfs \
        meta target-role=Started
primitive net_res ocf:redhat:ip.sh \
        params address=192.168.11.240 prefer_interface=bond0 \
        meta is-managed=true \
        meta target-role=Started
primitive nfsserv systemd:nfs-kernel-server \
```

```
        op monitor interval=30s \
        meta target-role=Started
ms drbd_master_slave drbd_res \
        meta globally-unique=false clone-max=2 notify=true target-
role=Started
location master-on-NAS1 drbd_master_slave \
        rule $role=master 100: #uname eq nfscluster1
colocation nfs_colo inf: nfsserv net_res fs_res drbd_master_slave:Master
order nfs_order Mandatory: drbd_master_slave:promote fs_res:start
net_res:start nfsserv:start
property stonith-enabled=off
```

Aide-mémoire Corosync / Pacemaker / DRBD

Surveiller le statut du cluster corosync:

```
crm_mon
```

Migrer les ressources vers un autre nœud:

```
crm resource migrate rg_main <fqdn_node_name>
```

Mettre un nœud hors ligne (attention cela met l'autre nœud en mode "préféré" pour forcer la transition, qu'il faut penser à retirer ensuite).

```
crm node standby
crm node online
```

Démarrer et arrêter toutes les ressources (attention cela met le cluster complètement hors-ligne, sans migration!)

```
crm resource stop rg_main
crm resource start rg_main
```

Afficher la configuration :

```
crm configure show
```

Si certaines ressources sont bloquées (non gérées) en mode "FAILED", par exemple à cause de l'échec d'une action stop, on peut remettre à zéro l'état:

```
crm_resource -P
crm resource cleanup rg_main
```

Attention, cela peut déclencher une migration si la ressource bloquée l'empêchait. Assurez-vous que vous êtes prêt à la faire.

Surveiller le statut du cluster *corosync* avec le nombre d'échecs :

```
crm_mon --failcount
```

Afficher le statut :

```
crm status
```

Vérifier le statut de DRBD :

```
cat /proc/drbd
```

Nettoyage d'un état DRBD "split-brain"

Sur le nœud secondaire :

```
drbdadm disconnect main
drbdadm -- --discard-my-data connect main
```

sur le nœud primaire :

```
drbdadm disconnect main
drbdadm primary main
drbdadm connect main
```

Divers

Scheduler optimizing on large arrays (untested) :

```
echo deadline > /sys/block/sdb/queue/scheduler
echo 0 >  /sys/block/sdb/queue/iosched/front_merges
echo 150 > /sys/block/sdb/queue/iosched/read_expire
echo 1500 > /sys/block/sdb/queue/iosched/write_expire
```

Système parallèle PVFS2/OrangeFS

configuration initiale de PVFS2

Tout d'abord il faut de préférence utiliser un fichier */etc/hosts* commun aux noeuds du cluster, pour s'assurer que chaque machine connait bien toutes les autres. Ensuite après avoir installé les paquets *pvfs-base* et *pvfs-2.6.XX* procéder à la configuration sur un des noeuds avec *pvfs2-genconfig /etc/pvfs2-fs.conf*. Le cluster sera nommé *pvfs2-fs*. Répondez bien aux questions, et n'utilisez pas *localhost* comme nom de machine mais les noms réels.

Ensuite :

- copiez le fichier */etc/pvfs2-fs.conf* sur tous les noeuds du cluster
- initialisez sur chaque noeud l'espace de stockage : `pvfs2-server /etc/pvfs2-fs.conf -f`
- démarrez le service pvfs2-server normalement sur chaque noeud : `service pvfs2-server start`

configuration du client

Démarrez le service *pvfs2-client* avant le script *mountpvfs2.sh*. *mountpvfs2.sh* utilise /etc/pvfs2tab, dont la syntaxe est identique à celle de fstab :

```
tcp://melodica:3334/pvfs2-fs /mnt/pvfs2 pvfs2 defaults,noauto 0 0
```

8.Restauration depuis le rescue

booter sur le CD de préférence. pour réparer la partition système, faire

```
e2fsck-y /dev/sda1
```

- par contre il vaut sans doute mieux restaurer carrément depuis le "rescue" parce qu'il y a un gros risque de fichiers manquants :

```
mount /dev/sda1 /mnt/sda1
ls -l /mnt/sda1/lost+found
```

- S'il y a des fichiers dans lost+found, c'est qu'ils sont perdus, il vaut mieux restaurer le backup :

```
mount /dev/sda3 /mnt/sda3
```

- vérifier que le "rescue" est à jour (par exemple il doit dater du 2 mars):

```
ls -ld /mnt/sda3/rdiff-backup-data
```

S'il est bien à jour, tu peux restaurer à la version du 2 mars :

```
rdiff-backup -r "2021/3/2" /mnt/sda3/ mnt/sda1/
```

- Ensuite il vaut mieux réinstaller le bootloader au cas où:

```
umount /mnt/sda3
cd /mnt/sda1
chroot .
mount /proc
mount /mnt/rescue
grub-install /dev/sda
umount -a
exit
```

- et pour finir :

```
reboot
```

9.Remplacement des disques contrôleurs 3Ware

Le remplacement des disques doit de préférence s'effectuer uniquement lorsque les unités RAID sont marquées "OPTIMAL". Attention : les raid_cli version 1.x et 2.x n'affichent pas toutes les erreurs de disque. Il faut faire

```
raid_cli info <contrôleur>
```

pour avoir toutes les infos, en particulier les status d'erreur de disques SMART-ERROR, DEVICE-ERROR. Marche à suivre :

- SMART-ERROR : remplacer préventivement le disque, RMA constructeur.
- DEVICE-ERROR : analyser les logs (*/var/log/messages*).

erreur timeout:

```
May 11 01:12:08 storiq-c1-n1 kernel: 3w-9xxx: scsi6: AEN: ERROR
(0x04:0x0009): Drive timeout detected:port=10.
```

On peut remplacer le disque, mais en général il ne présente pas de défaut lors des tests. En fait le firmware 3Ware en version antérieure à 4.10.07 n'efface pas les erreurs transitoires. Par contre, si le contrôleur est en 4.10.07, alors un "DEVICE-ERROR" est une vraie erreur permanente.

erreur "sector repair" :

```
May  7 12:45:15 storiq-c1-n1 kernel: 3w-9xxx: scsi6: AEN: WARNING
(0x04:0x0023): Sector repair completed:port=10, LBA=0xAFC.
```

On peut conserver le disque, mais il finira en SMART-ERROR tôt ou tard. Plutôt tôt d'ailleurs.

10.Gestion de l'alimentation

Il y a deux options : soit on utilise un UPS avec port série ou USB, soit un UPS réseau. Les UPS réseau malheureusement ne sont pas supportés par "nut".

Onduleur série/USB en local

Pour les séries/USB on utilisera nut : `aptitude install nut`

On configure le fichier: `/etc/nut/ups.conf` :

```
[apc]
    driver = usbhid-ups
    port = auto
```

Le nom entre crochets est libre. Pour les drivers possibles et les ups supportés, voir : http://www.networkupstools.org/compat/stable.html Si l'onduleur est série, il faut que nut puisse y accéder, on modifiera les règles udev pour cela en créant `/etc/udev/rules.d/99_nut-serialups.rules`:

```
# /etc/udev/rules.d/99_nut-serialups.rules
KERNEL=="ttyS1", GROUP="nut"
```

Ensuite on applique les changements dans udev:

```
sudo udevadm control --reload_rules
sudo udevadm control trigger
```

puis on démarre nut :

```
$ sudo upsdrvctl start
```

qui répondra :

```
Network UPS Tools - UPS driver controller 2.2.2
Network UPS Tools: 0.29 USB communication driver - core 0.33 (2.2.2)

Using subdriver: APC HID 0.92
```

Onduleur série/USB en réseau

Il faut aussi configurer upsd et upsmon. upsd communique avec le pilote; upsmon communique avec upsd et éteint la machine. Plusieurs upsmon et donc plusieurs machines peuvent être commandées depuis un seul upsd et un seul onduleur. Un upsd peut recevoir des messages de plusieurs onduleurs. Créer le fichier de configuration `/etc/nut/upsd.conf`:

```
# /etc/nut/upsd.conf
ACL all 0.0.0.0/0
ACL localhost 127.0.0.1/32
ACCEPT localhost
REJECT all
```

Dans cette configuration upsd n'accepte que les connexions du pilote local. Ensuite il faut remplir `/etc/nut/upsd.users`:

```
# /etc/nut/upsd.users
[local_mon]
    password = spider77
```

```
allowfrom = localhost
upsmon master
```

On peut utiliser plusieurs utilisateurs pour différentes machines. Ensuite on configure upsmon via `/etc/nut/upsmon.conf`:

```
# /etc/nut/upsmon.conf
MONITOR apc@localhost 1 local_mon spider77 master
POWERDOWNFLAG /etc/killpower
SHUTDOWNCMD "/sbin/shutdown -h now"
```

"apc" est le nom de l'onduleur indiqué dans `/etc/nut/ups.conf` et le mot de passe est celui donné dans `/etc/nut/upsd.users`. Ces fichiers doivent avoir des permissions restreintes :

```
$ sudo chown root:nut /etc/nut/*
$ sudo chmod 640 /etc/nut/*
```

Enfin il ne faut pas oublier d'activer nut au démarrage via `/etc/default/nut`:

```
# /etc/default/nut
START_UPSD=yes
START_UPSMON=yes
```

On démarre bien sûr avec

```
nut start
```

La commande suivante permet d'avoir un status de l'onduleur:

```
$ upsc apc
```

Attention, nut par défaut ne démarre l'extinction que quand l'onduleur est "critique", afin de ne pas couper sur une brève interruption.

Onduleur APC avec interface réseau

Les onduleurs APC avec interface réseau utilisent snmp et doivent être supervisés avec `apcupsd`. Il faut une version 3.10 pour supporter ces onduleurs.

Pour configurer l'adresse IP de l'onduleur, on peut lui attribuer une adresse via arp ainsi :

```
arp -s <IPaddress> <MacAddress>
ping <IPaddress> -s 113
```

Ensuite on se connecte sur l'onduleur en telnet, appuyer sur entrée 4 ou 5 fois, utilisateur "apc" mot de passe "apc",puis on peut enregistrer la configuration:

```
boot -b manual
tcpip -i <addresse>
tcpip -s <masque>
tcpip -g <routeur>
logout
```

ou via le menu, selon le type d'onduleur. Ensuite on configure le démon `apcupsd` via `/etc/apcupsd.conf`:

```
DEVICE 192.168.100.2:161:APC:private
```

Où les directives sont:

- addresse IP de l'onduleur
- port: port SNMP distant, normalement 161
- type d'agent SNMP:
  - "APC" pour les APC PowerNet?
  - "MIB", "APC_NOTRAP" pour la MIB PowerNet? avec les traps SNMP désactivés. (APC_NOTRAP nécessite apcupsd 3.12 ou supérieur).
- la communauté, normalement "private".

11.Économie d'énergie

installer cpufreq-utils ajouter dans /etc/modules powernow-k8 (noyau < 3.7) ou acpi-cpufreq (noyau > 3.7) et redémarrer.

12.Virtualisation avec lib-virt et virt-manager

Le but de la manœuvre : créer une interface bridge comme interface par défaut, qui sera automatiquement utilisé par virt-manager.

On doit donc créer une interface br0 qui intègre de base le groupe bond0 (pour continuer à utiliser le bonding).

Il suffit de modifier le fichier `/etc/network/interfaces` comme ceci:

1. on remplace "bond0" par "br0" partout.
2. on ajoute une ligne "bridge_ports bond0" dans les informations de br0
3. on ajoute une entrée "iface bond0 inet manual"

Ce qui donne ceci (avec vos ip à vous):

```
auto lo br0
iface lo inet loopback

iface bond0 inet manual

iface br0 inet static
        address 10.0.1.9
        netmask 255.255.0.0
        network 10.0.1.0
        broadcast 10.0.1.255
        gateway 10.0.1.1
        # paramètre bridge obligatoire
        bridge_ports bond0
        # paramètres bridge optionnels
        bridge_fd 2
        bridge_maxwait 1
```

Le redémarrage du réseau (service networking restart) n'a pas suffit chez moi. J'ai du faire comme ceci :

```
service networking stop
bonding_cli stop bond0
bonding_cli start bond0
service networking start
```

Par précaution, vérifiez que ça fonctionne bien en redémarrant le serveur. Au démarrage, on doit bien avoir un bridge br0.

Dès qu'il y a un bridge actif, virt-manager va automatiquement l'utiliser et insérer des interfaces virtuelles comme nécessaire, ça marche tout seul.

Pour pouvoir lancer virt-manager avec un utilisateur normal (pas root), il faut ajouter l'utilisateur en question au groupe libvirt:

```
adduser <user> libvirt
```

À partir de là toute nouvelle session de l'utilisateur en question pourra lancer le virt-manager.

Attention: les modes de bonding utilisant le monitoring arp ne fonctionneront pas correctement avec un bridge. Il faut soit désactiver le monitoring arp, soit utiliser le 802.3ad. Voir https://bugzilla.redhat.com/show_bug.cgi?id=584872#c14

13.Surveillance et supervision

SNMP

Le P.E.N. Intellique est 37990. La MIB SNMP intégrée à StorIQ 1.x et 2.x n'est pas valide et utilise un PEN de test. L'OID correct doit donc être "1.3.6.1.4.1.37990".

Pour sonder un service:

```
snmpwalk -c public -v1 d242
```

Pour sonder une MIB précise (particulièrement la notre...):

```
snmpwalk -v 1 -c public d242 .1.3.6.1.4.1.37990
```

Pour connaître la description des éléments de la MIB:

```
snmptranslate -IR -Tp -On storiq
```

Pour ajouter une MIB aux MIBS par défaut:

```
export MIBS=+STORIQ-TABLE-MIB
```